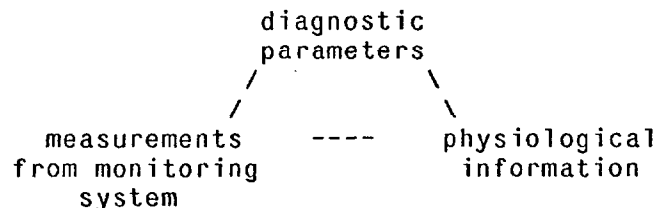## C. Recovery from Execution Failure

Once a problem solver observes a failure, there are a variety of
actions it can take. If adequate monitoring has been performed, the source
of the failure should be immediately evident. If not, the program must
localize the problem in order to correct it.

Once the failure is diagnosed, several responses are possible. The
problem solver may scrap its efforts and start from scratch; or it may pick
up from the failed state and produce another plan. In some cases, the
original plan may be used but only after the effects of the failed attempt
are undone. We would like to explore the techniques for dealing with
failure in a variety of settings from the laboratory to the computer
system, and we would like to study the trade-offs involved where several
approaches are possible.

6.3.1.3   Causal Models

    Medical and scientific reasoning depend on exploiting causal
relationships. We have encoded causal knowledge in production rules and
other representations without separating it from empirical associations.
This was a successful pragmatic approach. However, we recognize the
importance of representing and manipulating causal models as a separate
kind of knowledge in our reasoning programs, knowledge acquisition systems
and tutoring programs. We propose using the VM program as a test-bed and
point of takeoff for this research.

    The VM program provides real-time interpretation of the clinical
significance of measured data in the ICU. The project has considered the
relation between three related sets of abstract clinical data:
physiological information, measurements provided by a monitoring system,
and diagnostic parameters used in patient care. For example, VM uses a
patient's respiratory rate, a measured parameter, in interpreting his
effort of breathing, a diagnostic parameter. VM now includes a limited set
of physiological parameters, those directly related to measured data.
Associations between measured, diagnostic parameters and physiology are now
represented in VM when the association between these parameters is close
and apparent.

```
                         diagnostic
                         parameters
                       /            \
                      /              \
          measurements      ----      physiological
          from monitoring             information
               system
```

    The proposed research will increase the number of physiological
parameters used in the system and increase the number of interactions among
the three kinds of parameters which are represented in the system.
Specifically, this research will attempt to represent causal relations
among physiological parameters (based on a physiological model) and among
measurement parameters (based on a model of instrument function).

6.3.2   Knowledge Utilization and Tools for Building Expert Systems

6.3.2.1   Attempt to Generalize (AGE)

     The AGE system is currently supported under SUMEX core research and
its progress and future plans are described in Section 9.1.1 on page
137.

6.3.2.2   AI Handbook

     The AI Handbook is also supported currently under SUMEX core research
and its progress and future plans are described in Section 9.1.2 on page
145.

6.3.2.3   Research in Automated Consultation about Expert Systems

One of the drawbacks of knowledge based systems is that they are
often difficult to use. Consider, for example, a scientist trying to solve
a problem with a computer system he does not fully understand, and assume
that he has encountered a problem due to his lack of knowledge of that
system (say MACSYMA or MOLGEN). For example, he may be unaware of the
capabilities available, not know the system's vocabulary, or he might get
an answer he didn't expect. The simplest way for him to acquire just the
information he needs is to ask a consultant for help. Consultation is a
method widely used in computer centers; and, as complex programs become
more pervasive and more complex, the need for consultants will grow.
Unfortunately, consultants are scarce, expensive, and often unavailable
when needed.

One partial solution to this dilemma is in the form of automated
consultation about the use of complex programs. Recent work by Genesereth
in building an automated consultant for MACSYMA demonstrates the
feasibility of the approach, but there are many problems to be solved
before such consultants are put into general use.

Genesereth's program deals primarily with a user's violated
expectations about a system and tries to uncover and correct the
misconception responsible for those expectations. It assumes that the
user's actions are rational, i.e. that he has some plan for achieving his
goal. This plan explains why he chose the operations he did in terms of
his beliefs about those operations. The key to the identification of the
user's misconception is the recognition and debugging of this plan. In the
years ahead, we propose to extend and develop this "plan recognition"
approach and apply it to some of the computer capabilities available in the
SUMEX resource.

The importance of automated consultation should not be overlooked.
In general, a consultant is necessary whenever one is faced with a problem
solving situation in a domain one does not fully understand. The lack of
knowledge may be incidental, as it is when the domain is fairly simple but
time constraints make it impossible for the individual to learn all that is
necessary (e.g. with a simple text editor). Or, it may be essential, as
when the domain is very complex and the user can't possibly learn
everything (e.g. chemistry or MACSYMA).

6.3.2.4   <u>EMYCIN</u>

EMYCIN is a tool for building consultation systems within a backward-chaining framework. For small domains in which experts' judgmental knowledge is expressible in conditional rules, EMYCIN can provide rapid feedback on the adequacy of a rule set for providing reliable consultations. The INTERLISP version of EMYCIN is ready for use by others now. Future work includes the following:
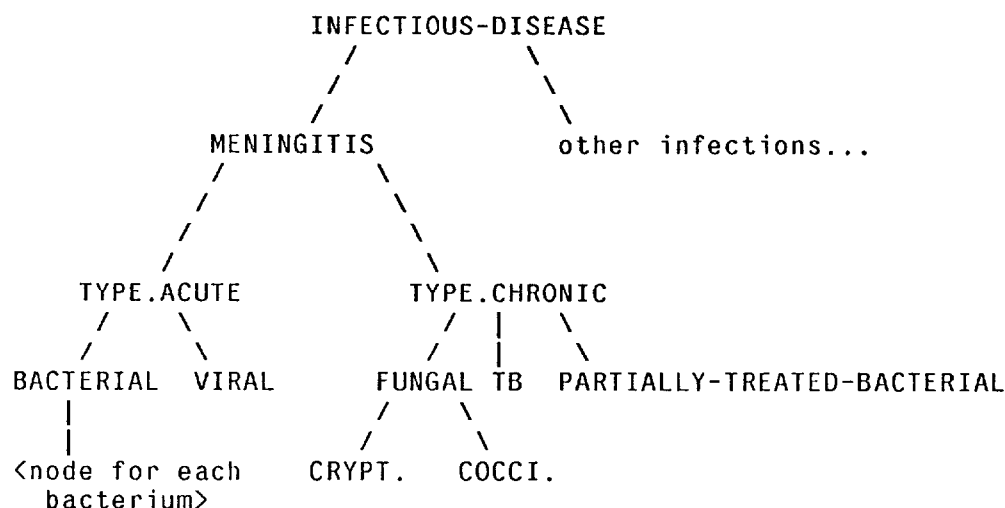
(1) Translation of EMYCIN for broader export

(2) Incorporation of strategic and structural information to integrate the information needed for tutoring and for acquiring new knowledge

The development of GUIDON, a case method tutor for EMYCIN knowledge bases, has given us a new perspective on the nature of the expertise that we have captured in our programs, and suggests guidelines for both representation and acquisition of knowledge. In particular, we have found that rules conveniently separate relationships into readily accessible associations, but an adequate knowledge base for teaching and acquisition requires the addition of structural knowledge (clusters and patterns), support knowledge (underlying causal mechanisms), and strategical knowledge (managerial approaches).

The strategical model is expressed in terms of rules in which the goal or action part is a task to carry out and the premise part consists of steps for achieving the goal. The strategical model will provide the foundation for a new version of EMYCIN which will encourage EMYCIN clients to incorporate in their specialized consultation systems the knowledge we have found to be useful for teaching.

We believe the strategical rules will be useful for controlling inferences as well as for teaching. Implementation of this idea requires that MYCIN's rule interpreter be modified slightly to recognize that some rules describe "tasks" that may be done repetitively, unlike "inference rules" which it only considers once for any case. With the addition of structural knowledge described below, MYCIN's backward-chaining interpreter can then be used to do hypothesis formation with focusing and non-exhaustive search.

Structural knowledge consists of clusters and patterns of rules and parameters--distinctions made by the strategical rules in their control of diagnostic reasoning. The central form of structural knowledge will be a taxonomic classification of the problem space. In MYCIN this will take the form of parameters that are hierarchically related to one another and share properties. One portion of the classification is shown below.

```
                    INFECTIOUS-DISEASE
                   /                 \
                  /                   \
                 /                     \
           MENINGITIS              other infections...
           /        \
          /          \
         /            \
        /              \
   TYPE.ACUTE        TYPE.CHRONIC
     /    \           /   |   \
    /      \         /    |    \
BACTERIAL  VIRAL   FUNGAL TB  PARTIALLY-TREATED-BACTERIAL
   |                /  \
   |               /    \
<node for each   CRYPT.  COCCI.
  bacterium>
```

Forming this classification involves regrouping the existing rule set,
creating a new parameter for each node in the hierarchy. This design of
taxonomic organization and inheritance of properties will make MYCIN's
representation more "frame-like," while preserving the use of rules to make
judgmental associations among the parameters.

     Because the strategical rules embody a weak model of diagnostic
behavior, we believe that they constitute a backbone that will be useful
for multiple problem areas. In particular, the strategical backbone could
be used to structure a knowledge acquisition dialogue. In addition to
encouraging a taxonomic classification of parameters, the strategical rules
indicate what other kinds of knowledge the expert building an EMYCIN system
will have to specify. For example, it is important to detail the knowledge
that suggests a broad category of problems that merit attention in a
particular case ("triggering associations") and knowledge to adequately
discriminate a case on the basis of the taxonomic distinction.

     Representing the diagnostic and strategical knowledge in a uniform
formalism of rules and parameters, and using an accepted backbone of
strategical knowledge, will enable us to use GUIDON for teaching from any
new EMYCIN-based program without needing to reorganize the consultation
knowledge base. The teaching program will be able to teach a student how
to approach cases, while the consultation program will direct its problem-
solving according the same approach, one that might be more acceptable to
physicians because it is patterned after their methods for solving
problems.

6.3.3   Knowledge Acquisition
Our research on knowledge acquisition to date has largely focused on adding
new knowledge to an existing knowledge base. A long-term effort is
proposed in which we focus on acquiring the structure and contents of a
whole knowledge base.

    The keystone of our approach to knowledge acquisition is the belief
that there is a substantial overlap in the knowledge of many different task
domains. We are not referring here to superficial facts and rules (say of
the physical world) but rather to the abstract structure implicit in even
quite disparate domains. For example, the notion of a hierarchy is found
in biological taxonomy, the classification of geologic time, and business
organization charts. The advantage of recognizing such abstract structures
is that they often possess efficient representations and efficient
algorithms for reasoning about them.

    In the past this commonality has not been exploited. One reason is
the difficulty of representing these abstract structures in a form directly
useable in different domains. Another problem is the difficulty of finding
and piecing together the structures appropriate to a novel domain. We
believe there is an elegant solution for these problems via the notions of
abstraction and simulation structure described below, and we propose to
develop a library of useful abstractions together with their specialized
representations and algorithms from which a knowledge engineer can pick and
choose in assembling expert programs.

    More specifically, we propose to expend our effort in four major
directions: (1) encoding useful abstractions and simulation structures, (2)
exploring the use of abstractions in checking the consistency and
completeness of knowledge bases, (3) automated selection of simulation
structures, (4) the use of abstractions in understanding analogy and the
use of analogies in identifying abstractions.

(1) A Library of Abstractions and Simulation Structures

    There are an infinite number of possible abstractions. What
motivates us to talk of a finite library is the fact that certain
abstractions have data representations or algorithms that are particularly
efficient or powerful. Some examples are trees, partial orders, rings,
groups, and monoids. We propose to differentiate simulation structures on
the basis of their representational economy and deductive power. For some
structures, this economy and power outweighs the uniformity of semantic
networks and frames. We intend to include only those abstractions for
which this is the case.

    A certain amount of theoretical work must precede the construction of
this library. We must devise an adequate language for describing
simulation structures and develop data and algorithm representations that
facilitates their interface and direct application in new domains. The
recent work on abstract operations by Barton, Genesereth, Moses, and Zippel
should help in this effort:

(2) The Use of Abstractions in Checking Consistency and Completeness

An abstraction prescribes a set of axioms that must be satisfied by all its models. These axioms can be used to check the consistency and completeness of the assertions a knowledge engineer makes in describing his task domain. For example, if a knowledge representation system suspected that a group of assertions was intended to describe a hierarchy, it could detect inconsistent data, such as cycles or multiple parents, and incomplete data, such as nodes without parents.

The abstractions appropriate to the task domain are determinable from a number of sources. The user may directly name the abstraction or describe it with an analogy; or the system may be able to infer it from partial information.

(3) Modeling

The use of models is a time-renowned problem solving technique. For example, architects and ship builders use models to get answers that would be too difficult to obtain using purely formal methods. We would like to draw an analogy between the architect's use of a physical model and the expert system's use of a simulation structure. In both cases the advantages to be gained are power and efficiency in reasoning about their domains.

Most knowledge representation systems store assertions in a uniform, domain-independent formalism like predicate calculus or semantic networks or frames. While there are advantages to uniformity and domain independence, these representations are in many cases considerably less efficient than specialized data structures, and the associated algorithms are often less efficient and less powerful. We are proposing to develop a systematic way of describing when well-known data representations and well-known algorithms are applicable and to devise a program able to employ simulation structures automatically in representing knowledge, given the abstractions it satisfies.

(4) Analogies

Many analogies are best understood as statements that the situations being compared share a common abstraction. For example, when one asserts that the organization chart of a corporation is like Linnaean taxonomy, what he is saying is that they are both hierarchies.

This view of analogy can be turned around and used to help novice users of our abstraction library in finding appropriate entries. Imagine an engineer describing the classification of time in geology (epochs, eras, periods, etc.) who can tell the system that his knowledge base is like that of biological taxonomy and have it infer and use the hierarchy abstraction.

In order to realize this goal, a number of problems must first be solved. The fundamental problem is completing a partial interpretation of

an abstraction. Once we have a method for completing interpretations,
analogy understanding (or at least the bit of it we are considering)
becomes easy. The system merely checks each of the abstractions of the
comparison domain, testing to see whether it is applicable.

Sometimes the system may not have a suitable prestored abstraction,
and this process will fail. Understanding an analogy in this situation
requires the invention of a new abstraction. We are interested in applying
and extending the concept formation techniques of Hayes-Roth, Mitchell, and
Dieterrich and Michalski in building a program to formulate new
abstractions automatically. Of course, a new abstraction will not
initially have any specialized data structures or algorithms, but it can
provide the next system builder with the techniques developed by the
originator.

6.3.4   Explanation

Our motivation for making explanation a primary focus of our research is a belief that expert systems will not be accepted by physicians or scientists unless the systems are able to justify the decisions they make. When important real world domains are involved, human decision makers are loathe to consult machines unless they understand and agree with the basis for the advice. This constraint not only forces us to consider mechanisms for generation of explanations, but it also impacts on the design of the underlying reasoning and representation techniques used by the rest of the consultation system.

In the case of MYCIN and its descendents, we have been able to generate intelligible explanations by taking advantage of our rule-based representation. Rules can be translated into English for display to a user, and their interactions can also be explicitly demonstrated. By adding mechanisms for understanding questions expressed in simple English, we were able to create an interactive system that allowed physicians to convince themselves that they agreed with the basis for the program's recommendations. MYCIN's explanation capabilities have been thoroughly discussed elsewhere [26].

MYCIN's explanation capabilities were generalized in EMYCIN and thus became available for any EMYCIN consultation system. They were further modified and utilized in both TEIRESIAS and GUIDON. Although we had experienced problems using MYCIN's rules for certain kinds of explanations (e.g., control mechanisms that were sometimes encoded in rules, or algorithmic knowledge such as the mechanisms for drug selection), it was in the setting of GUIDON that the inadequacies of MYCIN's approach became most apparent. Consider, for example, a simple MYCIN rule such as:

    If: the patient is less than 8 years old
    Then: don't give tetracycline

This rule is totally adequate for MYCIN's decision making task, and would be understood by most physicians if it were used in an explanation, but it is obvious to a casual observer that it contains a giant leap in logic. It is accordingly difficult for GUIDON to teach this rule to a novice medical student because the underlying pathophysiologic knowledge (i.e., that tetracycline is deposited in the developing bone and teeth of youngsters, weakening the former and disfiguring the latter) is not explicitly represented in MYCIN. Examples such as this one emphasize that a variety of knowledge forms are necessary if an intelligent system is to customize its explanations to the individual who is using the program. Underlying structural and causal relationships are generally required in addition to the high level judgmental rules that had contained almost all of the domain knowledge in MYCIN and the other EMYCIN systems.

During the second half on 1979 we formed a weekly seminar group to analyze the characteristics of good explanations. We generally tried to keep our discussions separate from computer science issues, concentrating instead on the psychology of explanation and planning to return eventually

to consider ways in which our developing theory might be implemented in
knowledge-based consultation systems. Although there are several
subproblems, it was agreed that the problems of explanation can generally
be divided into four categories: (1) modeling the knowledge of the system
user; (2) selecting a response strategy; (3) modeling contextual
information regarding the interaction; and (4) understanding the question.
One goal of our proposed work, then, is to build an explanation system
which explicitly addresses all four of these topics. We shall briefly
discuss each point:

(1) Modeling the User's Knowledge:

        GUIDON and other ICAI systems have recognized the need to keep an
internal model of the student, i.e., what he has shown he knows, what you
have already told him, and perhaps a record of where his greatest
weaknesses lie. Similarly, it is clear than an expert human consultant
customizes his explanations so that they can be understood by the person
requesting the consultation (and are thereby maximally convincing). The
expert starts with certain suppositions about his client's knowledge (e.g.,
a teacher may presume his student is starting from scratch, but a
cardiologist will assume that another physician requesting advice probably
already knows a fair amount of cardiology). The default presumption is
modulated, however, as the interaction proceeds and the client demonstrates
his strengths or weaknesses.

        We have recently begun some experiments to investigate methods for
encoding, along with the domain knowledge, the complexity and importance of
that knowledge. These two parameters seem to be independently important in
deciding whether to include a given reasoning step in an explanation.
"Key" points (i.e., those that are highly important) probably should be
mentioned even if they are not complex and are likely to be known to the
user. On the other hand, less important but complex items probably need
not be mentioned unless an expert user is really pressing for details of a
decision pathway. Thus, static measures of complexity and importance can
be compared with user descriptors that are initially assigned by default
(depending upon the status of the user, e.g., expert vs. student), but are
later altered dynamically in response to the course of the dialog and what
it has revealed about the user's background knowledge.

        These ideas have been encoded in a small computer program which uses
a limited knowledge base of rules and associations from the domain of
pharyngitis (sore throats). We have experimented with a semantic network
representation in which the nodes are values of attributes and rules are
only one form of link between nodes. All nodes and rules have complexity
and importance measures associated with them. An "opinion" regarding a
specific patient can be represented as a subset of the nodes in the
network, plus the links between them that account for how it has been
determined which nodes are active. In this setting, a question tends to
ask how it has been determined that a given node is active for a given
patient. The appropriate explanation could be very complex if an effort
were made to explain every link leading from data observations to the node
descriptor in question. A customized explanation is therefore generated

based on three variables which can be dynamically manipulated by the
program: (1) the focus of the dialog (e.g., broad-based vs. localized), (2)
the expertise of the user, and (3) the degree of generality which is
appropriate. These three variables are clearly not independent, and we are
experimenting with ways to have their values manipulated in a reasonable
fashion as the dialog proceeds.

This early effort will provide the basis for further discussions in
Year 1 of the proposed work. We have been fortunate to enlist the
collaboration of an endocrinologist at Stanford, Dr. Larry Crapo, who is
eager to work with us on building an endocrinology knowledge base. It is
likely that we will select the pathophysiology of thyroid disease, or of
the pituitary adrenal axis. Both these domains are appealing for computer-
based representation because the relationships are well-understood and
there are some challenging problems of feedback homeostasis that will need
to be represented. During Year 02 we will encode this knowledge base in
detail and begin experiments on the generation of explanations using the
kinds of techniques outlined above.

(2) Selecting A Response Strategy:

Our explanation efforts to date have tended to be simple reiterations
of individual reasoning steps, but it is clear that experts and teachers
use several alternate strategies for conveying their ideas or key facts.
Many of these techniques draw upon common sense world knowledge (e.g.,
analogies with familiar concepts outside the domain), but we have thus far
failed to capitalize on these teaching strategies in our work. Thus
another goal of the work that lies ahead will be to develop structures for
drawing parallels or otherwise representing the strategies used by good
"explainers."

(3) Modeling Contextual Information Regarding the Interaction

We have already mentioned some of the ways in which contextual
information may be useful in determining the best way to answer a question.
For example, a more accurate model of the user's knowledge can be developed
over time, and the extent to which a given conversation is focused on a
particular local topic can be assessed. Note that we are emphasizing here
issues other than those related to natural language understanding;
computational linguists also often cite the need to record contextual ·
dialog information in order to handle problems such as anaphora. An
understanding of the "flow" of a dialog is also important in understanding
the meaning of subsequent questions, as we discuss below.

(4) Understanding The Question

This issue interfaces with the problem of natural language
understanding, but we view it in a somewhat different light. We emphasize
instead the ways in which the model of the user and contextual information
may allow us to disambiguate questions. To draw from a medical example

that we have frequently discussed, consider the following scenario. A
reasoning program for pharyngitis diagnosis and management has just
diagnosed strep throat and recommended penicillin and the user asks the
question "Why would you give penicillin?" In the most obvious case, one
might imagine a response that itemizes the risks of streptococcal
infections and the reasons for treating early with penicillin. Similarly,
one might expect a more detailed response for a student and a quick summary
for a physician using the system.

However, an alternate interpretation is that EVERY physician knows
the theoretical reasons for giving penicillin in strep pharyngitis, and
that if the user is a physician and is asking the question then he must be
asking something different than the simple informational question. In this
case the query might be interpreted as a challenge (one that might have
been conveyed by tone of voice if it had been asked of a human consultant).
Apparently the user has reason to doubt that penicillin was the appropriate
agent in this case, or thinks that no drug was required. Other background
information and contextual knowledge should also help, and an intelligent
program might thereby answer the question in a given case in any of the
following ways:

"Because the patient has pre-existing rheumatic heart disease."

"Because I doubt that he is allergic to penicillin, even though he
reported that he is."

"Because he is unreliable and I am afraid I will not be able to reach
him to call him back if his strep culture comes back positive."

"Because I tend to treat conservatively and give penicillin for strep
throat even though I know there hasn't been a case of rheumatic heart
disease in California in over 10 years."

Note how different these kinds of explanations are from the simple
justification that a program such as MYCIN might have given:

"Because streptococcal pharyngitis may be followed by rheumatic
myocarditis or glomerulonephritis, mediated by immune complexes, and I
can prevent this complication by giving penicillin (to which
streptococci are uniformly sensitive)."

The ideal intelligent assistant should be able to determine from
knowledge of the user, the domain, the individual case, and the context of
the dialog, which of the preceding responses is most appropriate. We will
attempt to identify methods for giving our program this kind of capability.

7   Available Facilities

The existing SUMEX-AIM computer and communications facilities have
been described in earlier sections. The number of personnel to support
this follow-on work will remain at approximately the same level as before
so no additional office space will be required. The additional equipment
(VAX's, file server, and PWS's) will be accommodated in the existing SUMEX
machine room, a portion of the Pine Hall machine room allocated to Prof.
Feigenbaum, and in existing individual office areas. Technician support
and hardware development for this equipment will be housed in the existing
SUMEX electronics laboratory.

8   Literature Cited

1.   Feigenbaum, E.A., The Art of Artificial Intelligence: Themes and Case
     Studies of Knowledge Engineering, Proceedings of the 1978 National
     Computer Conference, AFIPS Press, (1978).

2.   Nilsson, N.J., Principles of Artificial Intelligence, Tioga Publishing
     Company, Palo Alto, California (1980).

3.   Winston, P.H., Artificial Intelligence, Addison-Wesley Publishing Co.,
     (1977).

4.   Nilsson, N.J., Artificial Intelligence, Information Processing 74,
     North-Holland Pub. Co. (1975).

5.   Barr A. and Feigenbaum, E.A. (Eds.), The Handbook of Artificial
     Intelligence, Stanford University Department of Computer Science,
     forthcoming.

6.   Boden, M., Artificial Intelligence and Natural Man, Basic Books, New
     York, (1977).

7.   McCorduck, P., Machines Who Think, W.H. Freeman and Co., San Francisco
     (1979).

8.   Coulter, C. L., Research Instrument Sharing, Science, Vol. 201, No.
     4354, August 4, 1978.

9.   Stefik, M., An Examination of a Frame-Structured Representation
     System, Proceeding of the Sixth International Joint Conference on
     Artificial Intelligence, Vol. 2, 845, August 1979.

10. Metcalfe, R.M. and Boggs, D.R., Ethernet: Distributed Packet Switching
     for Local Computer Networks, Comm. ACM, Vol. 19, No. 7 (July 1976).

11. Shoch, J.F. and Hupp, J.A., Performance of an Ethernet Local Network
     -- A Preliminary Report, Proceedings of the Local Area Communications
     Network Symposium, Boston,May 1979.

12. Taft, E.A., <u>Implementation of PUP in TENEX</u>, Internal XEROX PARC memorandum, June 1978.

13. Boggs, D.R., Shoch, J.F., Taft, E.A., and Metcalfe, R.M., <u>PUP: An Internetwork Architecture</u>, XEROX PARC report CSL-79-10, July 1979.

14. Wilcox, C. R., Jirak, G. A., and Dageforde, M. L., <u>MAINSAIL - Language Manual,</u> Stanford University Computer Science Report STAN-CS-80-791 (1980).

15. Wilcox, C. R., Jirak, G. A., and Dageforde, M. L., <u>MAINSAIL - Implementation Overview,</u> Stanford University Computer Science Report STAN-CS-80-792 (1980).

16. Mead, C. and Conway, L., <u>Introduction to VLSI Systems</u>, Addison-Wesley Publishing Co. (1980).

17. Rosen, B., <u>PERQ: A Commercially Available Personal Scientific Computer</u>, COMPCON 1980.

18. Thacker, C. P., McCreight, E. M., Lampson, B. W., Sproull, R. F., and Boggs, D. R., <u>ALTO: A Personal Computer</u>, Computer Structures: Readings and Examples (Siewiorek, Bell, and Newell, eds.), 1979.

19. McDaniel, <u>The Dorado: A Compact High-Performance Personal Computer for Computer Scientists</u>, COMPCON 1980.

20. Greenblatt, R., <u>MIT's LISP Machine</u>, COMPCON 1980.

21. Ward, S. and Terman, C., <u>An Approach to Personal Computing</u>, COMPCON 1980.

22. Lenat, D., "<u>AM: An Artificial Intelligence Approach to Discovery in Mathematics as Heuristic Search</u>, Ph.D. Dissertation, Stanford University, July 1976.

23. Stefik, M.J., <u>Planning With Constraints</u>, Ph.D. dissertation, Stanford University, January 1980.

24. Friedland, P.E., <u>Knowledge-Based Experiment Design In Molecular Genetics</u>, Ph.D. dissertation, Stanford University, October 1979.

Literature Cited

25. Sacerdoti, E.D., <u>Problem</u> <u>Solving</u> <u>Tactics</u>, Invited Lecture, Proceedings
     of the Sixth International Joint Conference on Artificial
     Intelligence, IJCAI-79. Available from Computer Science Dept.,
     Stanford University, August, 1979.

26. Scott, A.C., Clancey, W., Davis, R., and Shortliffe, E.H.,<u>Explanation</u>
     <u>Capabilities</u> <u>of</u> <u>Knowledge-Based</u> <u>Production</u> <u>Systems</u>, American Journal
     of Computational Linguistics, Microfiche 62, Knowledge-Based
     Consultation Systems, 1977.

Biographical Sketches

The following are biographical sketches for all professional
personnel contributing to the SUMEX-AIM resource project. These do not
include sketches for any of the individual collaborating project
investigators.

BIOGRAPHICAL SKETCH

(Give the following information for all professional personnel listed on page 3, beginning with the Principal Investigator.
Use continuation pages and follow the same general format for each person.)

NAME	TITLE	BIRTHDATE *(Mo., Day, Yr.)*
ACHENBACH, Michael W.	System Programmer	August 2, 1952

PLACE OF BIRTH *(City, State, Country)*	PRESENT NATIONALITY *(If non-U.S. citizen, indicate kind of visa and expiration date)*	SEX
Los Angeles, California, U.S.A.	U.S. Citizen	☒ Male ☐ Female

EDUCATION *(Begin with baccalaureate training and include postdoctoral)*

INSTITUTION AND LOCATION	DEGREE	YEAR CONFERRED	SCIENTIFIC FIELD
Stanford University	B.S.	1974	Physics
Stanford University	M.A.	1975	Education

HONORS

MAJOR RESEARCH INTEREST	ROLE IN PROPOSED PROJECT
Network communications, Small machines	System Programmer

RESEARCH SUPPORT *(See instructions)*

RESEARCH AND/OR PROFESSIONAL EXPERIENCE *(Starting with present position, list training and experience relevant to area of project. List all or most representative publications. Do not exceed 3 pages for each individual.)*

1978 – present	System Programmer, SUMEX Computer Project, Department of Genetics, Stanford University School of Medicine
1975 – 1978	Scientific Programmer, Instrumentation Research Laboratories, Department of Genetics, Stanford University School of Medicine
1975	Scientific Programmer, Institute for Mathematical Studies in the Social Sciences, Stanford University

PUBLICATIONS

Smith, D.H., Achenbach, M., Yeager, W.J., Anderson, P.J., Fitch, W.L., and Rindfleisch, T.C.: Quantitative Comparison Gas Chromatographic/ Mass Spectrometric Profiles of Complex Mixtures, Anal. Chem., 49, 1623, 1977.

BIOGRAPHICAL SKETCH

(Give the following information for all professional personnel listed on page 3, beginning with the Principal Investigator.
Use continuation pages and follow the same general format for each person.)

NAME	TITLE	BIRTHDATE *(Mo., Day, Yr.)*
AIELLO, Nelleke T.G.K.	Scientific Programmer	March 21, 1949

PLACE OF BIRTH *(City, State, Country)*	PRESENT NATIONALITY *(If non-U.S. citizen, indicate kind of visa and expiration date)*	SEX
Amsterdam, The Netherlands	U.S. Citizen	☐ Male ☒ Female

EDUCATION *(Begin with baccalaureate training and include postdoctoral)*

INSTITUTION AND LOCATION	DEGREE	YEAR CONFERRED	SCIENTIFIC FIELD
University of California, Santa Cruz	B.A.	1971	Mathematics
University of California, Santa Cruz	B.A.	1971	Information and Computer Science
University of Utah, Salt Lake City	M.S.	1972	Computer Science

HONORS

Departmental Honors, Information and Computer Science, University of California

Crown College Honors, University of California

MAJOR RESEARCH INTEREST	ROLE IN PROPOSED PROJECT
Building intelligent systems Knowledge engineering	Scientific Programmer

RESEARCH SUPPORT *(See instructions)*

RESEARCH AND/OR PROFESSIONAL EXPERIENCE *(Starting with present position, list training and experience relevant to area of project. List all or most representative publications. Do not exceed 3 pages for each individual.)*

1977 – present	Scientific Programmer, Heuristic Programming Project, Computer Science Department, Stanford University
1972 – 1977	Programmer, Bolt, Beranek and Newman, Inc.
1973 – 1975 Summers	Teaching Assistant, Structured Programming, University of California Extension
Summer 1972	Teaching Assistant, Compiler Writing, University of California Extension
1971	Programmer, Shell Benelux Centre, De Hage, The Netherlands

PUBLICATIONS   (See continuation page)

BIOGRAPHICAL SKETCH - AIELLO, Nelleke T.G.K.
PUBLICATIONS

1. Aiello, N: An Analysis of Notations for Music Applicable
    to the Digital Control of Electronic Musical Instruments.
    Masters Thesis, University of Utah, 1972.

2. Collins, A.M., Warnock, E.L., Aiello, N., and Miller, M.L.:
    Reasoning from Incomplete Knowledge, In D. Bobrow and
    A.M. Collins (Eds.) REPRESENTATION AND UNDERSTANDING STUDIES
    IN COGNITIVE SCIENCE. New York, Academic Press, Inc., 1975.

3. Nii, H.P. and Aiello, N.: AGE (Attempt to Generalize): Profile
    of the AGE-0 System. Stanford Heuristic Programming Project
    Memo HPP-78-5 (Working Paper), June 1978.

4. Nii, H.P. and Aiello, N.: AGE: A knowledge-based program for
    building knowledge-based programs. Proc. of IJCAI-6, pp 645-655,
    1979.

5. Aiello, N., Nii, H.P. and White, W.C.: The Joy of AGE-ing: An
    Introduction to AGE-1. Stanford Heuristic Programming Project Memo
    (work in progress), May 1980.

BIOGRAPHICAL SKETCH

(Give the following information for all professional personnel listed on page 3, beginning with the Principal Investigator.
Use continuation pages and follow the same general format for each person.)

NAME BUCHANAN, Bruce G.	TITLE Adjunct Professor Computer Science	BIRTHDATE *(Mo., Day, Yr.)* July 7, 1940
PLACE OF BIRTH *(City, State, Country)* St. Louis, Missouri, U.S.A.	PRESENT NATIONALITY *(If non-U.S. citizen, indicate kind of visa and expiration date)* U.S. Citizen	SEX ☒ Male ☐ Female

EDUCATION *(Begin with baccalaureate training and include postdoctoral)*

INSTITUTION AND LOCATION	DEGREE	YEAR CONFERRED	SCIENTIFIC FIELD
Ohio Wesleyan University	B.A.	1961	Mathematics
Michigan State University	M.A.	1966	Philosophy
Michigan State University	Ph.D.	1966	Philosophy

HONORS

(see continuation page)

MAJOR RESEARCH INTEREST Artificial Intelligence	ROLE IN PROPOSED PROJECT Technical Director of Core Research

RESEARCH SUPPORT *(See instructions)*

(see continuation page)

RESEARCH AND/OR PROFESSIONAL EXPERIENCE *(Starting with present position, list training and experience relevant to area of project. List all or most representative publications. Do not exceed 3 pages for each individual.)*

1976 – present   Adjunct Professor, Computer Science Department, Stanford University

1972 – 1976   Research Computer Scientist, Computer Science Department,
               Stanford University

1966 – 1971   Research Associate, Artificial Intelligence Project,
               Stanford University

PUBLICATIONS (see continuation page)

BIOGRAPHICAL SKETCH - BUCHANAN, Bruce G.

RECENT HONORS

Editorial Board, Artificial Intelligence: An International Journal
American Association for Artificial Intelligence - Organizing Committee,
        Program Committee and Membership Chairman
Chairman of Program Committee, IJCAI-79 (International Joint Conference
        on Artificial Intelligence, Tokyo, 1979)
Invited Colloquium Speaker:
        University of Maryland
        Carnegie-Mellon University
        Rutgers University
        University of California at Berkeley
        Michigan State University
Invited Speaker:
        AISB Annual Conference (Amsterdam, July 1980)
        Workshop on the Logic of Discovery and Diagnostics in Medicine
                (Pittsburgh, October 1978)
        Douglass College Seminars for Faculty (Rutgers University, 1978)
        Workshop on Pattern Directed Inference Systems (Honolulu, 1977)
Recipient, National Institutes of Health Career Development Award (1971-1976)

MEMBERSHIPS
        American Association for Artificial Intelligence (AAAI)
        Cognitive Science Society
        Association for Computing Machinery (ACM), SIGART
        Philosophy of Science Association

RESEARCH SUPPORT

| Grant No. | Title of Project | Funding | | % of Effort | Grant Agency |
		Current Year	Project Period		
1P01 LM 03395-01	Research Program : Biomedical Knowledge Representation	$ 99.484 (7/79-6/80)	$497,420 (7/79-6/84)	10	NLM
MCS-7903753	Knowledge-Based Consultation Systems	$ 73,659 (7/79-6/80)	$ 73,659 (7/79-6/80 + 6 months)	10	NSF
N00014-79-C-0302	Exploration of Tutoring and Prob. Solv. Strategies in Intelligent Computer-Aided Instruction	$396,325 (3/79-3/82)	$396,325 (3/79-3/82)	10	ONR
MDA 903-80-C-0107	Heuristic Programming Project	$496,256 (10/79-9/80)	$1,613,588 (10/79-9/82)	40	ARPA
5R24 RR00612-10	Resource-Related Research - Computers and Chemistry	$221,255 (5/80-4/81)	$641,419 (5/80-4/83)	5	NIH

Selected Publications

Edward H. Shortliffe, Bruce G. Buchanan, and Edward A. Feigenbaum, "Knowledge Engineering for Medical Decision Making: A Review of Computer-Based Clinical Decision Aids," Proceedings of the IEEE, September, 1979.

Bruce G. Buchanan, "Issues of Representation in Conveying the Scope and Limitations of Intelligent Assistant Programs." In J.E. Hayes, D. Michie, and L.I. Mikulich (eds.), Machine Intelligence 9: Machine expertise and the human interface. New York: John Wiley, 1979.

Bruce G. Buchanan and Edward A. Feigenbaum, "DENDRAL and Meta-DENDRAL: Their Applications Dimension," Artificial Intelligence 11, 5, 1978.

Bruce G. Buchanan, Tom M. Mitchell, Reid G. Smith and C. Richard Johnson, Dr., "Models of Learning Systems," in J. Belzer (ed.), Encyclopedia of Computer Sciences and Technology, New York: Marcel Dekker, Inc., 1978, Vol 11.

Randall Davis and Bruce G. Buchanan, "Meta-Level Knowledge: Overview and Applications," Proceedings of the Fifth IJCAI,1,920, August 1977.

Bruce G. Buchanan and Tom Mitchell. "Model-Directed Learning of Production Rules," in D.A. Waterman and F. Hayes-Roth (eds.), Pattern Directed Inference Systems, New York: Academic Press, 1978.

Bruce G. Buchanan and Dennis Smith, "Computer Assisted Chemical Reasoning," in E.V. Ludena, N.H. Sabelli and A.C. Wahl (eds.), Computers in Chemical Education and Research, New York: Plenum Press, 1977, p. 401.

Randall Davis, Bruce Buchanan, Edward Shortliffe, "Production Rules as a Representation of a Knowledge-Based Consultation Program," in Artificial Intelligence, 8, 1, February 1977.

Bruce G. Buchanan, D.H. Smith, W.C. White, R.J. Gritter, E. Feigenbaum, J. Lederberg, and C. Djerassi, "Application of Artificial Intelligence for Chemical Inference XXII. Automatic Rule Formation in Mass Spectronomy by Means of the Meta-DENDRAL Program," Journal of the American Chemical Society, 98, 6168, 1976.